

RRRRRRRRRRRR		UUU	UUU	NNN	NNN	000000000	FFFFFFFFFFFFFF	FFFFFFFFFFFFFF
RRRRRRRRRRRR		UUU	UUU	NNN	NNN	000000000	FFFFFFFFFFFFFF	FFFFFFFFFFFFFF
RRRRRRRRRRRR		UUU	UUU	NNN	NNN	000000000	FFFFFFFFFFFFFF	FFFFFFFFFFFFFF
RRR	RRR	UUU	UUU	NNN	NNN	000	FFF	FFF
RRR	RRR	UUU	UUU	NNN	NNN	000	FFF	FFF
RRR	RRR	UUU	UUU	NNN	NNN	000	FFF	FFF
RRR	RRR	UUU	UUU	NNNNNN	NNN	000	FFF	FFF
RRR	RRR	UUU	UUU	NNNNNN	NNN	000	FFF	FFF
RRR	RRR	UUU	UUU	NNNNNN	NNN	000	FFF	FFF
RRRRRRRRRRRR		UUU	UUU	NNN	NNN	000	FFFFFFFFFFFFFF	FFFFFFFFFFFFFF
RRRRRRRRRRRR		UUU	UUU	NNN	NNN	000	FFFFFFFFFFFFFF	FFFFFFFFFFFFFF
RRRRRRRRRRRR		UUU	UUU	NNN	NNN	000	FFFFFFFFFFFFFF	FFFFFFFFFFFFFF
RRR	RRR	UUU	UUU	NNN	NNNNNN	000	FFF	FFF
RRR	RRR	UUU	UUU	NNN	NNNNNN	000	FFF	FFF
RRR	RRR	UUU	UUU	NNN	NNNNNN	000	FFF	FFF
RRR	RRR	UUU	UUU	NNN	NNN	000	FFF	FFF
RRR	RRR	UUU	UUU	NNN	NNN	000	FFF	FFF
RRR	RRR	UUU	UUU	NNN	NNN	000	FFF	FFF
RRR	RRR	UUU	UUU	NNN	NNN	000	FFF	FFF
RRR	RRR	UUUUUUUUUUUUUUUU	NNN	NNN	000000000	FFF	FFF	
RRR	RRR	UUUUUUUUUUUUUUUU	NNN	NNN	000000000	FFF	FFF	
RRR	RRR	UUUUUUUUUUUUUUUU	NNN	NNN	000000000	FFF	FFF	

```
NN      NN  DDDDDDDD  XX      XX  PPPPPPPP  000000  LL
NN      NN  DDDDDDDD  XX      XX  PPPPPPPP  000000  LL
NN      NN  DD      DD  XX      XX  PP      PP  00      00  LL
NN      NN  DD      DD  XX      XX  PP      PP  00      00  LL
NNNN     NN  DD      DD      XX  XX  PP      PP  00      00  LL
NNNN     NN  DD      DD      XX  XX  PP      PP  00      00  LL
NN  NN     NN  DD      DD      XX  XX  PPPPPPPP  00      00  LL
NN  NN     NN  DD      DD      XX  XX  PPPPPPPP  00      00  LL
NN      NNNN  DD      DD      XX  XX  PP      PP  00      00  LL
NN      NNNN  DD      DD      XX  XX  PP      PP  00      00  LL
NN      NN  DD      DD  XX      XX  PP      PP  00      00  LL
NN      NN  DD      DD  XX      XX  PP      PP  00      00  LL
NN      NN  DDDDDDDD  XX      XX  PP      PP  000000  LLLLLLLLLL
NN      NN  DDDDDDDD  XX      XX  PP      PP  000000  LLLLLLLLLL
                                     ....
                                     ....
                                     ....
                                     ....
```

```
RRRRRRRR  EEEEEEEEE  QQQQQQ
RRRRRRRR  EEEEEEEEE  QQQQQQ
RR      RR  EE      QQ      QQ
RR      RR  EE      QQ      QQ
RR      RR  EE      QQ      QQ
RR      RR  EE      QQ      QQ
RRRRRRRR  EEEEEEEEE  QQ      QQ
RRRRRRRR  EEEEEEEEE  QQ      QQ
RR      RR  EE      QQ      QQ
RR      RR  EE      QQ      QQ
RR      RR  EE      QQ      QQ
RR      RR  EE      QQ      QQ
RR      RR  EEEEEEEEE  QQQQ  QQ
RR      RR  EEEEEEEEE  QQQQ  QQ
```

Version: 'V04-000'

* COPYRIGHT (c) 1978, 1980, 1982, 1984 BY *
* DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS. *
* ALL RIGHTS RESERVED. *

* THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED *
* ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE *
* INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER *
* COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY *
* OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY *
* TRANSFERRED. *

* THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE *
* AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT *
* CORPORATION. *

* DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS *
* SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL. *

++
FACILITY:

DSR (Digital Standard RUNOFF) /DSRPLUS DSRINDEX/INDEX Utility

ABSTRACT:

This file contains literals and macros defining the data structures
found in the internal index pool

ENVIRONMENT: Transportable

AUTHOR: JPK

CREATION DATE: January 1982

MODIFIED BY:

003 JPK00015 04-Feb-1983
Cleaned up module names, modified revision history to
conform with established standards. Updated copyright dates.

002 JPK00009 24-Jan-1983
Modified to enhance performance. The sort buckets have each
been divided into 27 sub-buckets; 1 for each letter and 1
for non-alphas. Removed reference to BUCKET from INDEX.
Definition of the structure was added to NDXPOL. References
to BUCKET were changed in modules NDXOUT, NDXINI, NDXFMT
and NDXDAT.

--

! Index entry

\$FIELD XE_FIELDS =
SET

XESA_PREV	=	[\$ADDRESS],	! Link to previous item
XESA_NEXT	=	[\$ADDRESS],	! Link to next item
XESA_SUBX	=	[\$ADDRESS],	! Sub index pointer
XESA_REF	=	[\$ADDRESS],	! Reference pointer
XESA_TEXT	=	[\$ADDRESS],	! Pointer to text of index item
XESA_SORT_AS	=	[\$ADDRESS],	! Pointer to SORT_AS string
XESH_SUBC	=	[\$SHORT_INTEGER],	! Sub index level

XESV_FLAGS	=	[\$SHORT_INTEGER],	! Entry flags
------------	---	--------------------	---------------

\$OVERLAY (XESV_FLAGS)

XESV_BARS	=	[\$BIT],	! Change bar flag
-----------	---	----------	-------------------

\$CONTINUE

XESA_BOOK_LIST	=	[\$ADDRESS]	! Master index book name list
----------------	---	-------------	-------------------------------

\$ALIGN (FULLWORD)

TES;

LITERAL

XESK_LENGTH = \$FIELD_SET_SIZE;

MACRO

\$XE_BLOCK = BLOCK [XESK_LENGTH] FIELD (XE_FIELDS) %;

! End of Index entry

! Reference entry

\$FIELD XX_FIELDS =
SET

XXSA_LINK	=	[\$ADDRESS],	! Link to additional entries
XXSA_APPEND	=	[\$ADDRESS],	! APPEND text pointer
XXSH_PAGE	=	[\$SHORT_INTEGER],	! Transaction number

XXSV_FLAGS	=	[\$SHORT_INTEGER],	! Display attributes
------------	---	--------------------	----------------------

\$OVERLAY (XXSV_FLAGS)

XXSV_BOLD	=	[\$BIT],	! Bold page reference
XXSV_UNDERLINE	=	[\$BIT],	! Underline page reference
XXSV_BEGIN	=	[\$BIT],	! Begin page range
XXSV_END	=	[\$BIT],	! End page range

\$CONTINUE

```
XXSA_BOOK          = [$ADDRESS]          ! Master index book name
$ALIGN (FULLWORD)
TES;
LITERAL
  XXSK_LENGTH = $FIELD_SET_SIZE;
MACRO
  $XX_BLOCK = BLOCK [XXSK_LENGTH] FIELD (XX_FIELDS) %;
! End of Reference entry

! Master index book reference entry
$FIELD XM_FIELDS =
  SET
  XMSA_LINK          = [$ADDRESS],        ! Link to additional entries
  XMSA_BOOK          = [$ADDRESS]        ! Pointer to book name
  TES;
LITERAL
  XMSK_LENGTH = $FIELD_SET_SIZE;
MACRO
  $XM_BLOCK = BLOCK [XMSK_LENGTH] FIELD (XM_FIELDS) %;
! End of Master index book reference entry

! Current Entry
$FIELD C_FIELDS =
  SET
  CSA_CURR           = [$ADDRESS],        ! Pointer to current cell
  CSA_PREV           = [$ADDRESS],        ! Pointer to previous cell
  CSA_HEAD           = [$ADDRESS],        ! Pointer to head of chain
  $ALIGN (FULLWORD)
  CSV_FLAGS          = [$INTEGER],        ! Current cell flags
  $OVERLAY (CSV_FLAGS)
  CSV_IDNS           = [$BIT]            ! Identical string flag
  $CONTINUE
  TES;
```

```
LITERAL
  CSK_LENGTH = $FIELD_SET_SIZE;
```

```
MACRO
  $C_BLOCK = BLOCK [CSK_LENGTH] FIELD (C_FIELDS) %;
```

```
! End of current entry
```

```
!
! Dummy datasets
```

```
LITERAL
  DS_X_ENTRY = X$K_LENGTH,
  DS_XX_ENTRY = XX$K_LENGTH,
  DS_XM_ENTRY = XM$K_LENGTH,
  DS_X_STRING = 0;
```

```
! Structure definition for bucket array.
```

Buckets are arranged so that each row represents the first letter of the string and each column represents the second letter of the string.

This approach is used only for master indexes as no performance improvement is realised until about 10 input files have been processed.

Indexes which are not master indexes use only the first element of each row, i.e., [0, 0] ... [26, 0].

The only exception is for nonalphabetic characters which use only element [0, 0]. Elements [0, 1] ... [0, 26] are not used since mapping all nonalphabetic characters into one row loses the sort order of the first character in the string. For nonalphabetic characters to work correctly in a two dimensional bucket scheme, the array would have to be at least 127 x 127

	0	1		26
0	**	not used	:	.
1	A?	AA	:	AZ
:			:	:
:			:	:
26	Z?	ZA	.	ZZ

```
STRUCTURE
  $BUCKET_ARRAY [ROW_IDX, COL_IDX; M, N] =
    [M * N * %UPVAL] ($BUCKET_ARRAY + (ROW_IDX * N + COL_IDX) * %UPVAL);
```

```
!-- End of NDXPOL.REQ
```


CONURT REQ			FRMSTK REQ	GETQSC REQ						NDXCLT REQ	NDXRTY REQ			
	ECC REQ							KWITEM REQ					PHDEF REQ	RUNTAB REQ
			FLIPRECS REQ	FNCT REQ		GNCC REQ	IFSTK REQ		LSTOPS REQ			OUTOPT REQ		
DIGIT REQ					F5PACK REQ						NDXXPL REQ			
									MAXIMA REQ				POOL REQ	RUNHAN REQ
DMDEFS REQ	FFDEFS REQ				GSLUCC REQ	INDEX REQ				NDXLIN REQ		PAGEN REQ		
							LETTER REQ				NMLST REQ			
			FOOFIL REQ	GCA REQ				MSG REQ					RNODEF REQ	
DSRLIB REQ	FLGT REQ				HCT REQ	IRAC REQ				NDXPOL REQ		PASS REQ		
							LODEFS REQ					NUMPRM REQ		
			FOOREC REQ					MSGTXT REQ	NBITS REQ					
						KC REQ						PDT REQ	RNOMAC REQ	
	FLIRCHRS REQ				HLC REQ		LSTBS REQ				OPDEV REQ			